Analiza punktów funkcyjnych

Miara wielkość funkcjonalnej oprogramowania

Tomasz Koszlajda
Instytut Informatyki Politechniki Poznańskiej

Potrzeby określenia wielkości oprogramowania

- Dane wejściowe dla estymacji nakładu pracy i czasu wykonania. Nakład pracy i czas wykonania są funkcją rozmiaru budowanego oprogramowania.
- Dane dla planowania harmonogramu i alokacji zasobów. Harmonogram i alokacja zasobów muszą być uzależnione od rozmiaru budowanego oprogramowania.
- Określanie produktywności zespołu i poszczególnych pracowników – poprzez określenie rozmiaru oprogramowania zbudowanego w jednostce czasu.
- Dane dla określenia stanu zaawansowania procesu projektowania. Pozwala określić procentowo stan zaimplementowanej funkcjonalności.
- Dane dla procesu utrzymania oprogramowania. Pozwala oszacować rozmiar modyfikacji w kontekście całego oprogramowania i w konsekwencji koszty modyfikacji.
- Dane gromadzone dla kolejnych projektów.

Paradoks produktywności

Linie kodu źródłowego

Wielkość źródłowego kodu programu jest mierzona za pomocą liczby fizycznych lub logicznych linii programu źródłowego (ang. source line of code – SLOC), zazwyczaj z pominięciem linii pustych i komentarzy.

Charakterystyka:

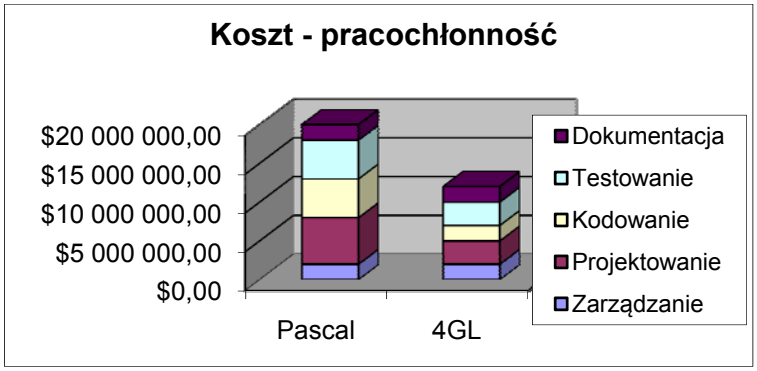
- prostota i dokładność
- obiektywność metody pomiarowej
- zależność od narzędzi programowych
- zależność od wykonawców (od jakości wykonania)
- możliwość pomiaru dopiero po etapie kodowania
- szacowanie rozmiaru kodu źródłowego
  - **Wideband-Delphi**

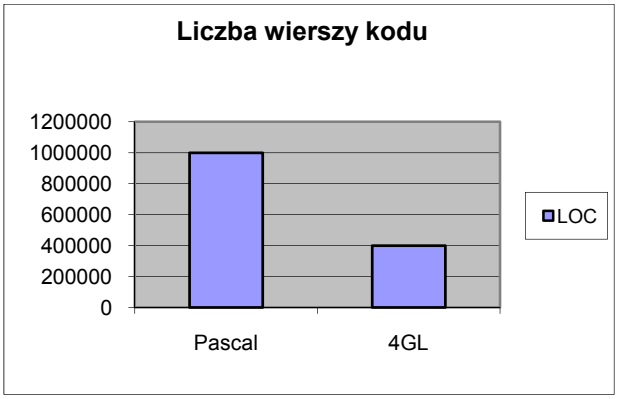
    Opiera się na współpracy grupy ekspertów
  - **Logiki rozmytej**

    Wymaga dużej liczby danych historycznych
  - **Standardowych komponentów**

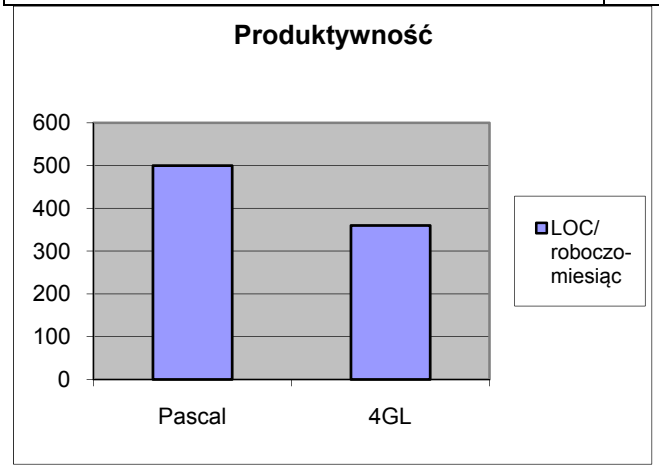
    Zakłada stałość rozmiarów standardowych elementów konstrukcyjnych
- **paradoks produktywności**

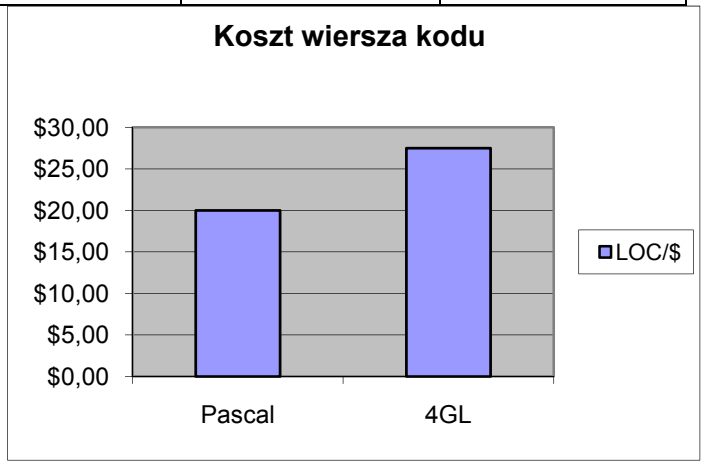
Paradoks produktywności





	Przypadek A	Przypadek B	Różnica
Koszt	Pascal 1 000 000 LOC	4GL 400 000 LOC	
Specyfikacja wymagań	2 000 000	2 000 000	0
Analiza i projekt	6 000 000	3 000 000	3 000 000
Kodowanie	5 000 000	2 000 000	3 000 000
Testowanie	5 000 000	3 000 000	2 000 000
Dokumentowanie	2 000 000	2 000 000	0
Nakład pracy [pm]	2 000	**1 110**	890
Koszt [$]	20 000 000	**12 000 000**	8 000 000
Produktywność [LOC/m]	500	**360**	**140**
Koszt jednostkowy [$/LOC]	20	**27,50**	**-7,50**





Analiza punktów funkcyjnych

Umożliwia pomiar funkcjonalnej złożoności programów. Mierzona jest uniwersalna funkcjonalność związana z przetwarzaniem danych. Proces pomiaru wymaga identyfikacji, klasyfikacji i szacowania wielkości elementarnych funkcjonalności.

Albrecht IBM 1979, 1983

Caspar Jones (70) – paradoks produktywności

Alternatywne metody miar funkcjonalności

- **IFPUG v4.3.1 (2010)**
- **MKII v1.3.1 (1998)**
- **NESMA v2.2 (2004)**
- **COSMIC FFP v3 (2007)**

Powyższe propozycje są zdefiniowane jako standardy ISO

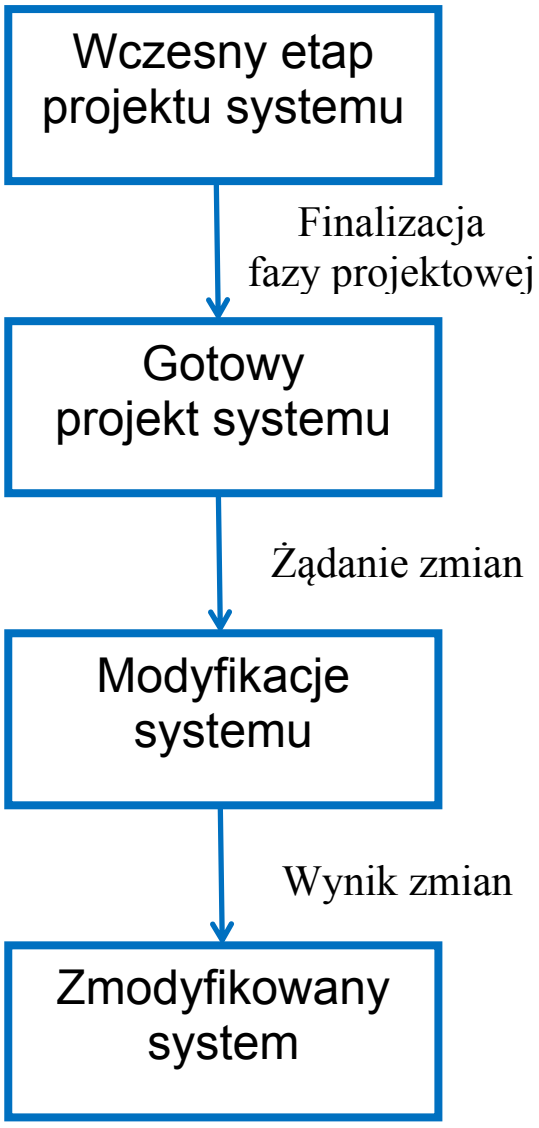
Charakterystyka punktów funkcyjnych

- **Niezależność miary od technologii informatycznych**: stosowanych języków programowania, architektury programów i metodyk procesu projektowania.
- **Niezależność miary od jakości wykonania**, polegającej na uzyskaniu tego samego efektu różnymi sposobami i rozmiarami implementacji.
- **Możliwość zastosowania miary we wszystkich etapach projektu**: strategia, analiza, projekt, implementacja, utrzymanie. W ramach strategii można dokonać oszacowania rozmiaru, kolejne etapu pozwalają na dokładne wyliczenie rozmiaru.
- **Niezależność miary od nakładu pracy** potrzebnego do implementacji danej funkcjonalności, dzięki temu stanowią punkt odniesienia dla oceny różnych narzędzi implementacyjnych.
- **Miara stosowalna w ograniczonej dziedzinie zastosowań**, głównie systemów biznesowych. Gorzej sprawdza się w zastosowaniach systemowych, czasu rzeczywistego, itp.

Klasy analizy punktów funkcyjnych

- Analiza funkcjonalności **projektowanych systemów informatycznych** (ang. ***development project function point count***) – służy do wyznaczania rozmiaru funkcjonalności dostarczanej użytkownikom z pierwszą instalacją systemu informatycznego.
- Analiza funkcjonalności **modyfikacji systemów informatycznych** (ang. - ***enhancement project function point count***) mierzy rozmiar funkcjonalny modyfikacji istniejących systemów informatycznych. Modyfikacje obejmują dodawanie, zmianę i usuwanie dostępnej funkcjonalności.
- Analiza funkcjonalności **eksploatowanych systemów informatycznych** (ang. ***application function point count***) – dotyczy eksploatowanego systemu informatycznego. Punkty funkcyjne eksploatowanych systemów informatycznych są inicjowane przez liczbę punktów funkcyjnych wdrożonego systemu i modyfikowane w wyniku rozbudowywania systemu.

Klasy analizy punktów funkcyjnych



Oszacowanie FP budowanego systemu

(1) Analiza FP budowanego systemu

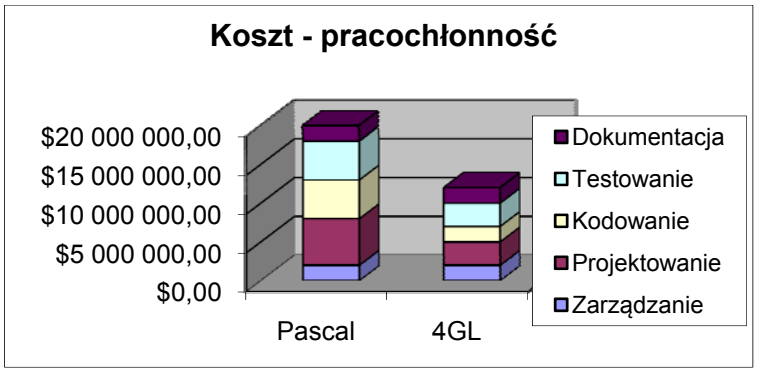
+

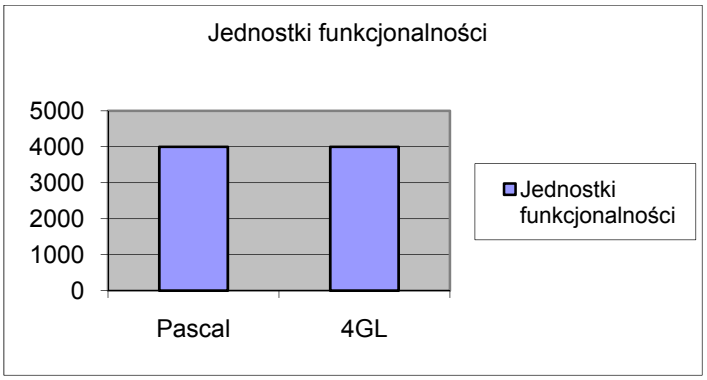
(2) Analiza FP modyfikacji systemu

≠

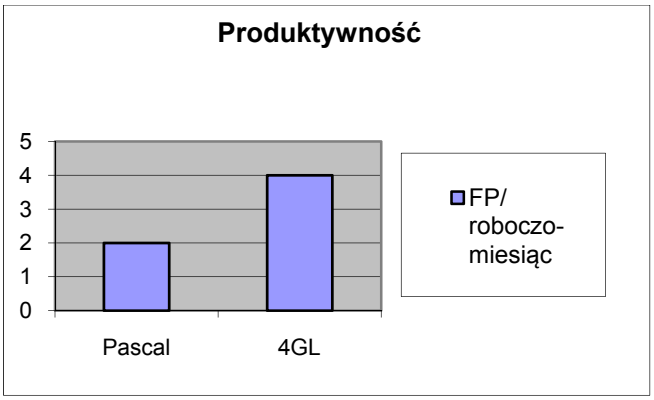
(3) Analiza FP zmodyfikowanego systemu

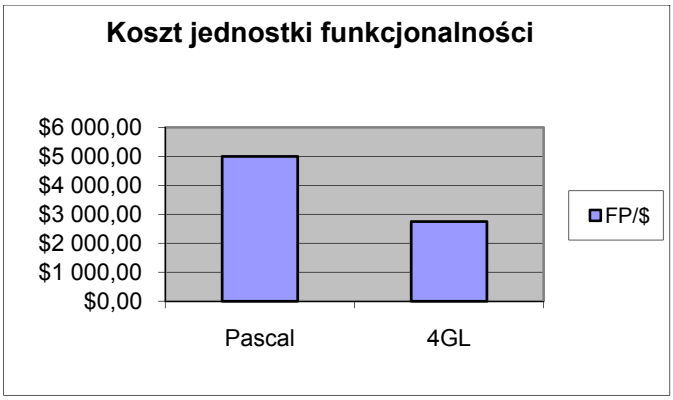
Produktywność mierzona w punktach funkcyjnych



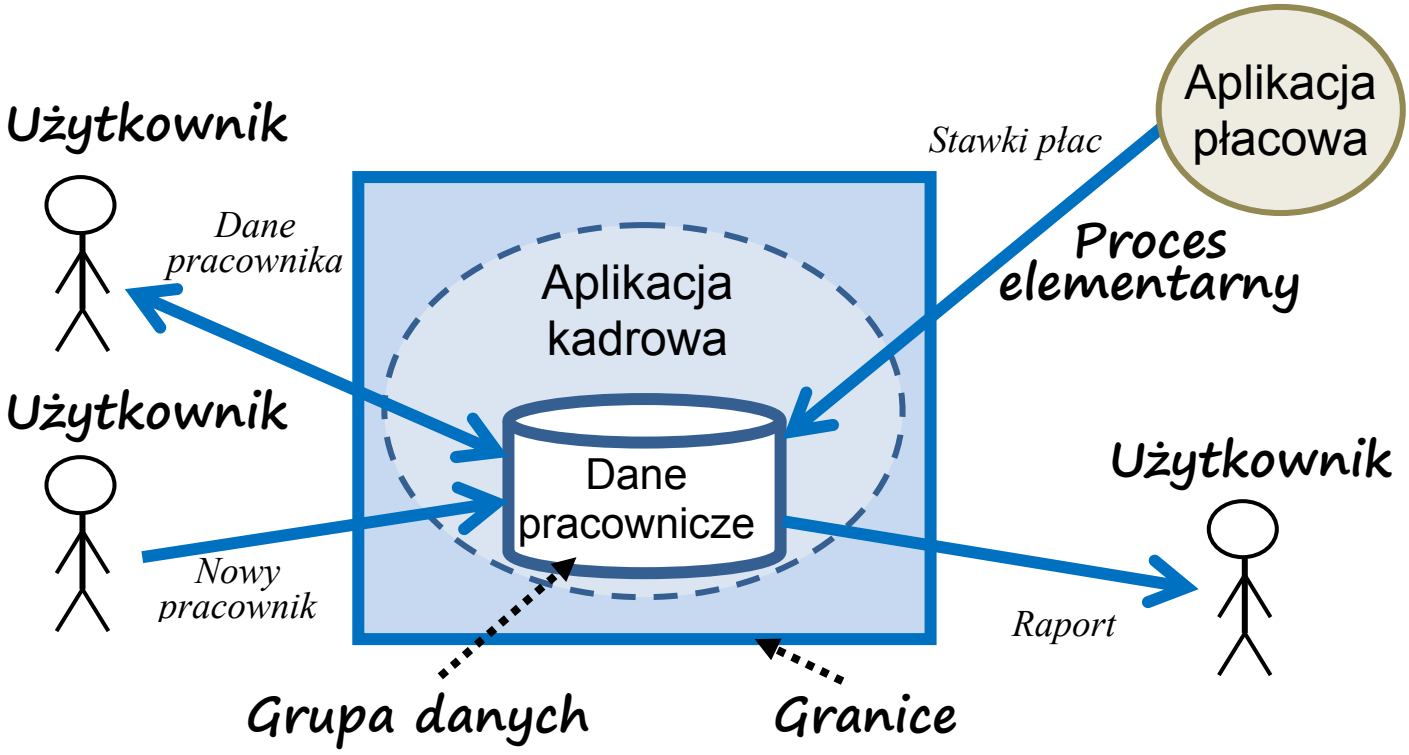


	Przypadek A	Przypadek B	Różnica
Koszt	Pascal	4GL	
Funkcjonalność	4 000	4 000	0
Nakład pracy [pm]	2 000	**1 110**	890
Koszt [$]	20 000 000	**12 000 000**	8 000 000
Produktywność [FP/m]	2	**4**	-2
Koszt jednostkowy [$/FP]	5 000	**2 750**	2 250





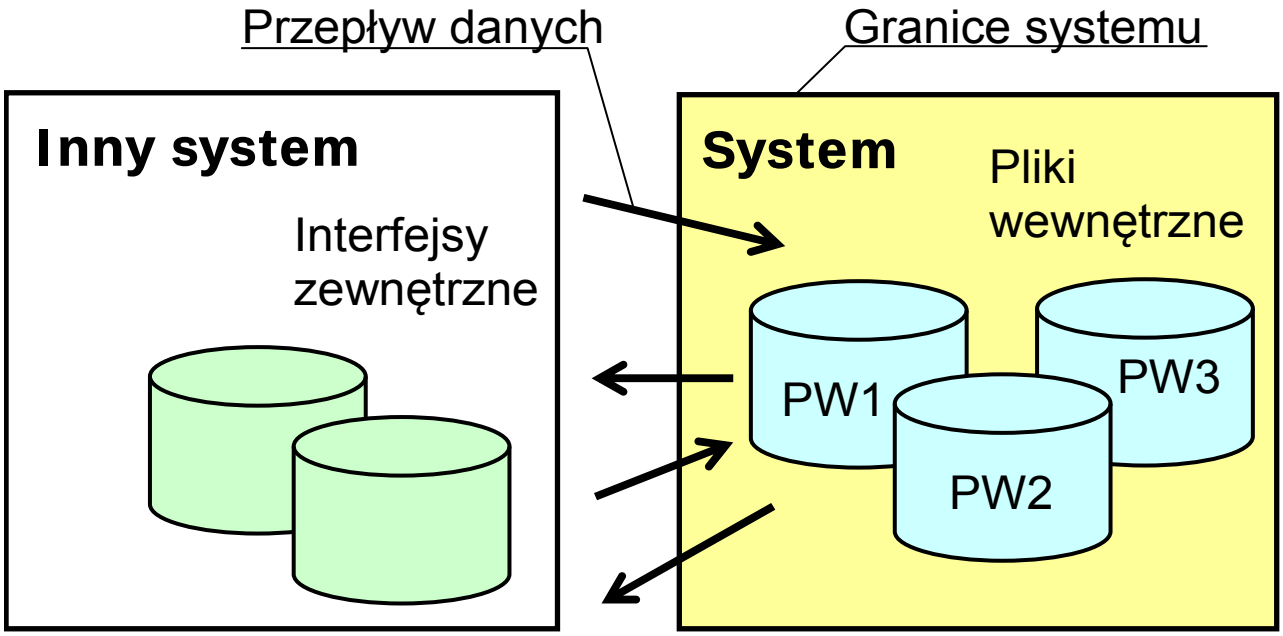
Model oprogramowania



- **Granice** – między mierzonym programem komputerowym lub projektem oprogramowania, a światem zewnętrznym: użytkownikami, innymi programami komputerowymi lub urządzeniami. Określa podzbiór funkcji których rozmiar ma być obliczany.
- **Użytkownicy** – osoba, program komputerowy lub urządzenie, które wchodzą w interakcję z mierzonym oprogramowaniem
- **Grupy danych** – dane identyfikowane i grupowane razem z perspektywy funkcjonalnej
- **Elementarne procesy** – najmniejsza jednostka aktywności znacząca dla użytkownika końcowego w procesie biznesowym

Klasy funkcjonalności systemów informatycznych IFPUG

Odwzorowanie specyfikacji systemów na abstrakcyjny model funkcjonalności



Pięć klas funkcjonalności

Przepływy danych – dane w ruchu

- **Wejścia danych** - **EI** (ang. external inputs)
- **Wyjścia danych** - **EO** (ang. external outputs)
- **Zapytania** - **EQ** (ang. external inquiries)

Logiczne pliki danych – dane składowane

- **Wewnętrzne pliki danych** – **ILF** (ang. internal logical files)
- **Zewnętrzne pliki danych** – **EIF** (ang. external interface files)

Wyznaczanie złożoności funkcjonalności

Dwupoziomowa klasyfikacja elementarnych funkcjonalności:

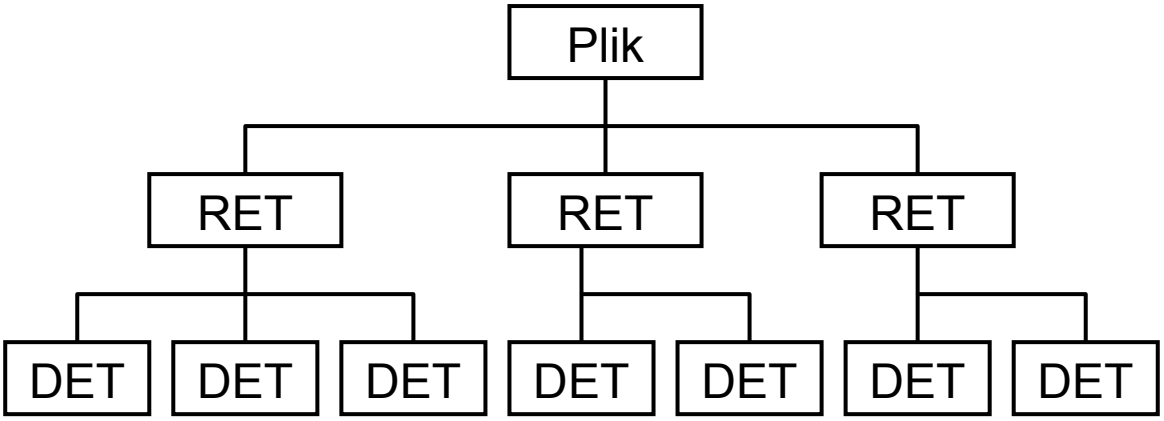
- Klasa funkcjonalności
- Złożoność funkcjonalności

Typ funk ji	Złożoność		
	Niska	Średnia	Wysoka
Plik wewnętrzny	7	10	15
Plik zewnętrzny	5	7	10
Wejście danych	3	4	6
Wyjście danych	4	5	7
Zapytanie	3	4	6

Reguły wyznaczania złożoności logicznych plików danych

Złożoność funkcjonalna danych składowanych w plikach logicznych jest wyznaczana na podstawie dwóch parametrów:

- Liczby różnych struktur danych **RET** (ang. Record Element Type) występujących wewnątrz danego pliku
- Liczby elementów danych **DET** (ang. Data Element Type) występujących w danym pliku danych



Elementy DET są rozróżnialnymi przez użytkowników, nie powtarzającymi się atrybutami danych.

Elementy RET są rozróżnialnymi przez użytkowników podzbiorami elementów danych w pliku.

Wskazówki do liczenia punktów funkcyjnych logicznych plików danych

- Logiczne pliki danych reprezentują dane istotne z punktu widzenia użytkowników systemów informatycznych. Nie obejmują one struktur danych, których istnienie wynika z wymagań implementacyjnych.
- Pojęcia plików danych ILF i EIF odpowiadają pojęciom: encji z diagramów encji-związków, relacji w schemacie relacyjnej bazy danych lub pliku w systemie plików danych.
- Nie każda relacja lub fizyczny plik muszą odpowiadać logicznym plikom danych ILF i EIF. Tabele lub pliki zawierające dane nieistotne lub niewidoczne z biznesowej perspektywy użytkownika nie są identyfikowane jako pliki logiczne ILF lub EIF.
- Encje, relacje lub fizyczne pliki nie są powiązane relacją 1:1 z plikami logicznymi ILF lub EIF. Plik logiczny może reprezentować wiele encji, relacji i plików fizycznych. Równocześnie jedna relacja lub plik fizyczny mogą być reprezentowane przez wiele plików logicznych. Dobry diagram encji-związków powinien gwarantować, że jedna encja będzie reprezentowane przez co najwyżej jeden plik logiczny.

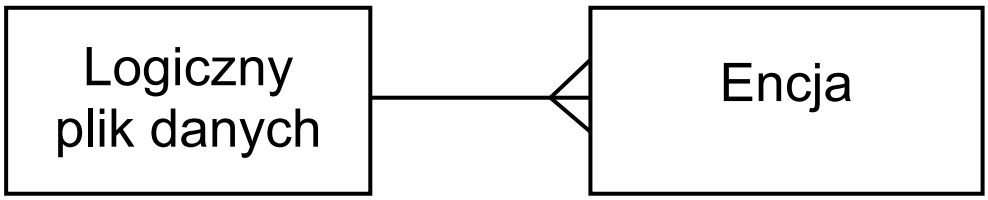
Wyznaczanie plików logicznych na podstawie ERD

Diagramy encji-związków reprezentują punkt widzenia danych przez użytkowników systemów informatycznych. W związku z tym, wszystkie encje występujące na diagramie powinny być brane pod uwagę przy identyfikacji i określaniu złożoności logicznych plików danych ILF i EIF.

Elementy DET odpowiadają wszystkim atrybutom encji oraz wszystkim związkom łączącym encje.

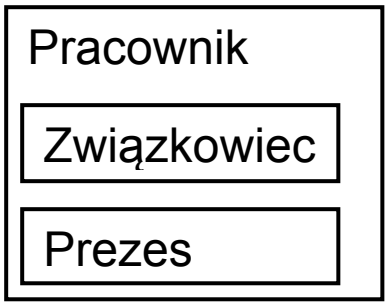
Pliki logiczne i encje tworzą związki typu:

- 1:1 – liczba RET w pliku jest równa jeden,
- 1:N – liczba RET w pliku jest większa od 1, złożone logiczne pliki danych.

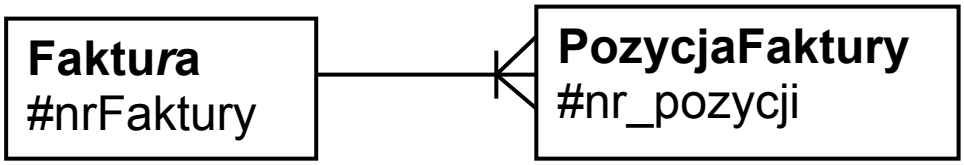


Złożone logiczne plik danych

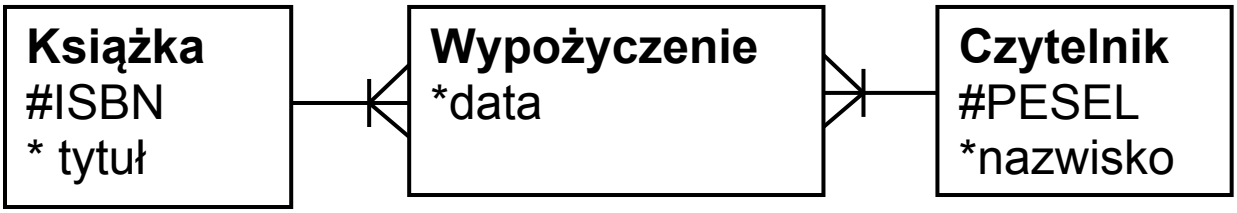
1. Hierarchie encji – pod-encje wchodzące w skład hierarchii encji nie tworzą nowych plików logicznych, tylko dodatkowe struktury danych wewnątrz pliku logicznego odpowiadającego encji będącej korzeniem hierarchii. Wyjątkiem mogą być pod-encje, których zbiory wystąpień są rozłączne i kompletne.



2. Encje słabe identyfikowane przez jeden związek, modelujące silny związek kompozycji między encjami. Encje słabe nie tworzą nowych plików logicznych, tylko dodatkowe struktury danych wewnątrz jednego pliku logicznego odpowiadającego encji silnej identyfikującej encję słabą.



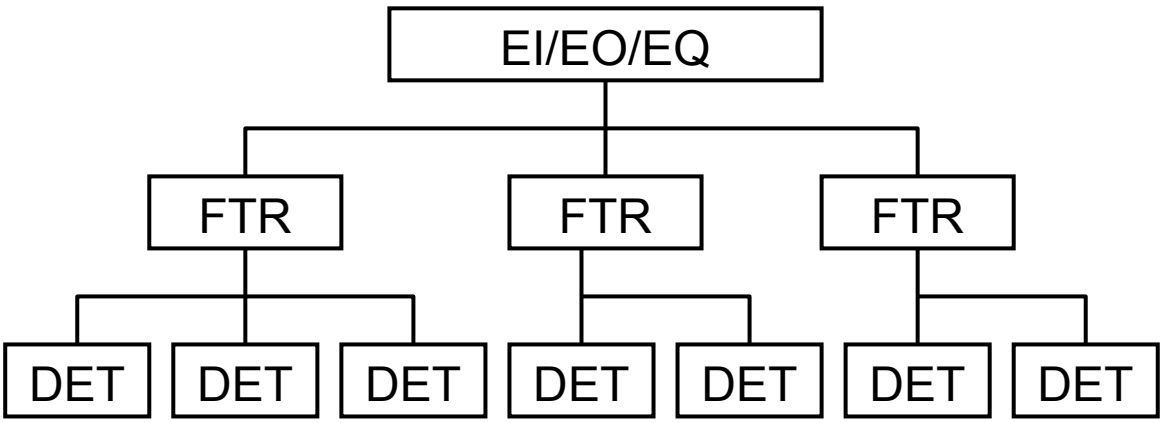
3. Encje słabe identyfikowane przez dwa lub więcej związków, modelujące encje połączeniowe (związki M:N z atrybutami). Encja słaba zostanie dodana jako dodatkowa struktura danych do obydwu plików.



Reguły wyznaczania złożoności przepływów danych

Złożoność funkcjonalna wejść i wyjść danych oraz zapytań jest szacowana na podstawie parametrów:

- Liczby przetwarzanych plików danych **FTR** (ang. File Type Referenced). Przetwarzanie obejmuje: odczyt, zapis, modyfikację i usuwanie danych w plikach logicznych.
- Liczby pól danych (elementów danych) **DET** (ang. Data Element Type) wpływających do systemu i wypływających z systemu.



Liczba DET jest sumą:

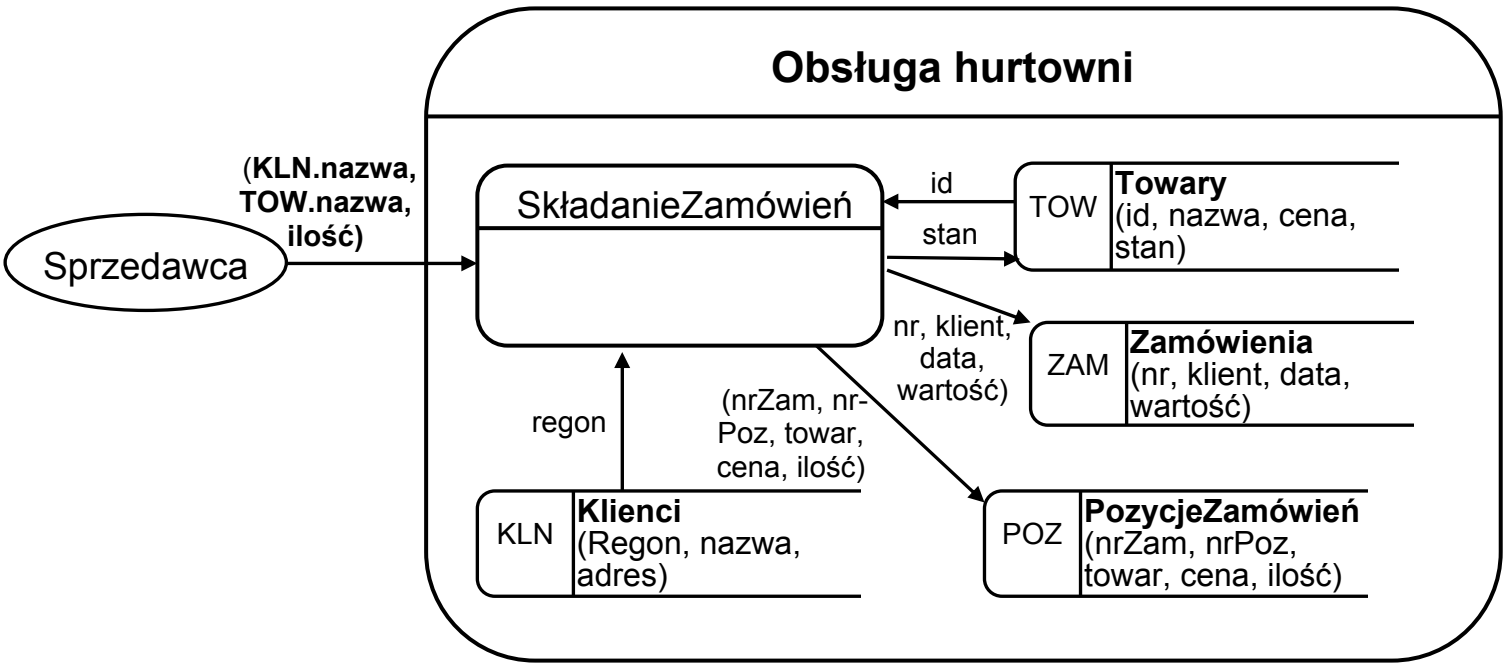
- **Wejść**: pola danych wejściowych (wstawianych, modyfikowanych i usuwanych), które będą składowane wewnątrz systemu, dane kontrolne, komunikaty o błędach.
- **Wyjść**: pola danych wyjściowych (odczytywanych, wyliczonych i modyfikowanych), dane kontrolne, np. komunikaty o błędach, nagłówki kolumn, itp.

Wskazówki do identyfikacji funkcjonalności przepływu danych

- Identyfikowane przepływy danych (transakcje) są minimalnymi przepływami z punktu widzenia logiki biznesowej zdefiniowanej przez użytkowników systemu.
- Pojedynczy ekran użytkownika, raport lub program wsadowy, może być reprezentowany przez wiele elementarnych przepływów danych.
- Wiele ekranów użytkownika, raportów lub programów wsadowych, może być reprezentowanych przez jeden elementarny przepływ danych.
- Określenie typu przepływu wynika z dominującego kierunku przepływu danych. Pojedynczy przepływ danych może jednak obejmować transfer danych w dwóch kierunkach. Na przykład, zapytanie może wymagać przepływu do systemu parametrów określających jego zakres.
- Jako przepływy liczone są tylko transfery danych przekraczające granice systemu. Przepływy danych zamykające się wewnątrz systemu nie wpływają na jego funkcjonalność, lecz określają złożoność jego implementacji.

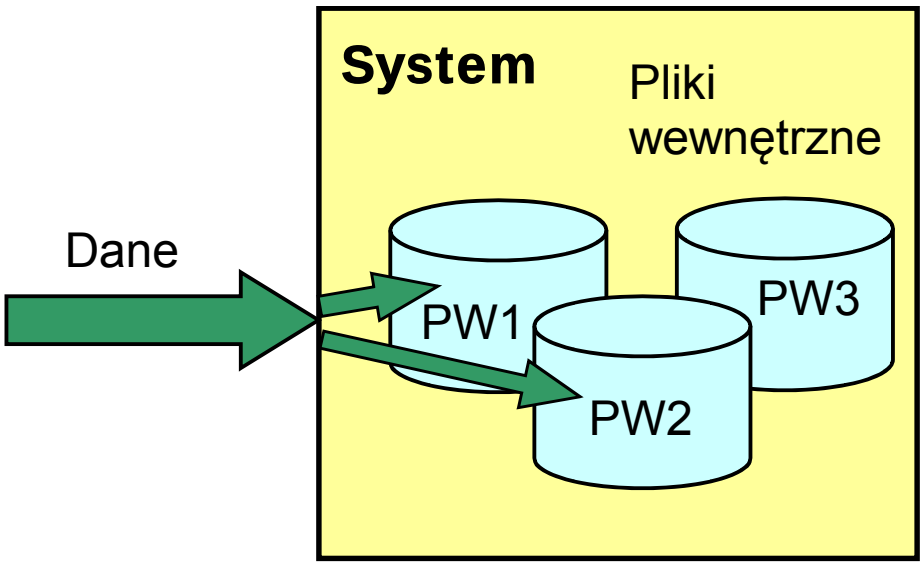
Wyznaczanie przepływów danych na podstawie diagramów DFD

- Diagramy przepływu danych reprezentujące przepływy danych między światem zewnętrznym, a modelowanym systemem informatycznym reprezentują punkt widzenia użytkowników systemów informatycznych i są podstawą do identyfikacji i szacowania złożoności programów komputerowych.
- Wszystkie funkcje realizujące przepływy danych do i z systemu powinny być zidentyfikowane jako elementarne przepływy danych.
- Zbiorniki danych (ang. data store) odczytywane i zapisywane przez te funkcje, są podstawą do określenia złożoności przepływu na podstawie liczby przetwarzanych logicznych plików danych.
- Atrybuty przypisane strzałkom przepływów danych są podstawą do określenia liczby elementów DET danego przepływu.



Wejścia danych

Wejście danych jest elementarnym przepływem danych, w którym dane przekraczają granicę systemu do jego wnętrza. Dane mogą być wprowadzane przez formatki ekranowe lub inne aplikacje. Dane mogą zawierać informacje merytoryczne lub kontrolne. Dane merytoryczne zasilają wewnętrzne pliki danych. Dane kontrolne mają wpływ na przebieg przetwarzania i nie muszą być składowane w plikach danych. Z wejściami wiążą się operacje wstawiania, modyfikacji i usuwania danych z plików danych.

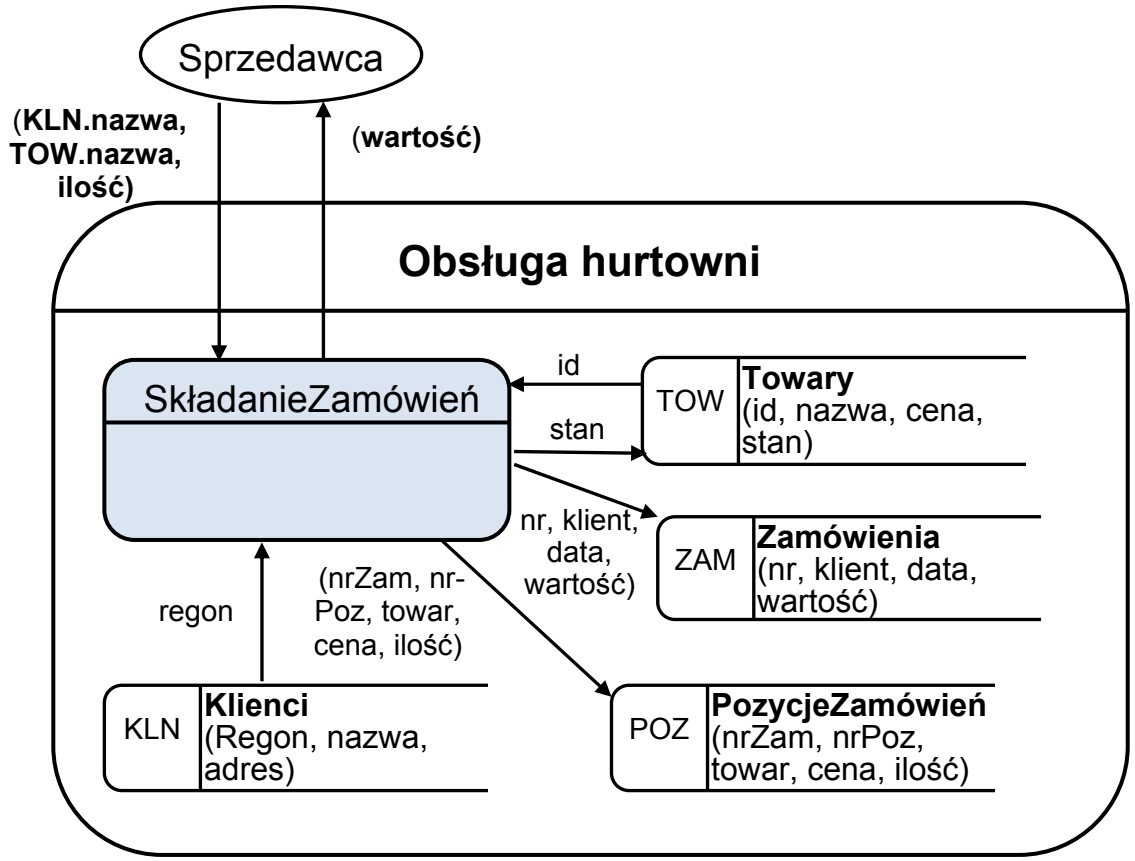


Wyznaczanie złożoności *wejść danych*.

FTR	DET		
	1-4	5-15	> 15
< 2	Niska(3)	Niska(3)	Średnia(4)
2	Niska(3)	Średnia(4)	Wysoka(6)
> 2	Średnia(4)	Wysoka(6)	Wysoka(6)

Wejścia danych

Przykład: funkcja wprowadzania danych o zamówieniach.



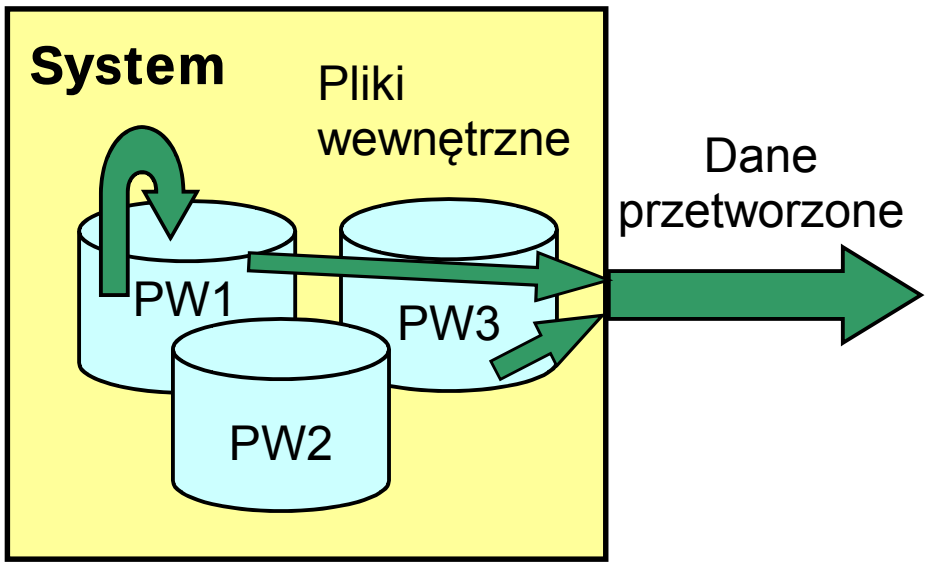
Typ funkcjonalności: wejście danych
FTR: 3 (TOW, ZAM+POZ, KLN)
DET: 4 (KLN.nazwa, TOW.nazwa, ilość, wartość)

Elementy danych: KLN.nazwa, TOW.nazwa mimo, że wprowadzane wielokrotnie dla tego samego zamówienia, są liczone tylko raz.

FTR	DET		
	1-4	5-15	> 15
< 2	Niska(3)	Niska(3)	Średnia(4)
2	Niska(3)	Średnia(4)	Wysoka(6)
> 2	Średnia(4)	Wysoka(6)	Wysoka(6)

Wyjścia danych

Wyjście danych jest elementarnym przepływem danych, w którym dane przekraczają granicę systemu na zewnątrz. Dodatkowo działanie to może wiązać się z modyfikacją wewnętrznych plików danych. Dane są wyprowadzane na zewnątrz w postaci raportów lub plików przekazywanych do innych aplikacji. Dane wynikowe powstają poprzez przetworzenie danych składowanych w wewnętrznych plikach danych.

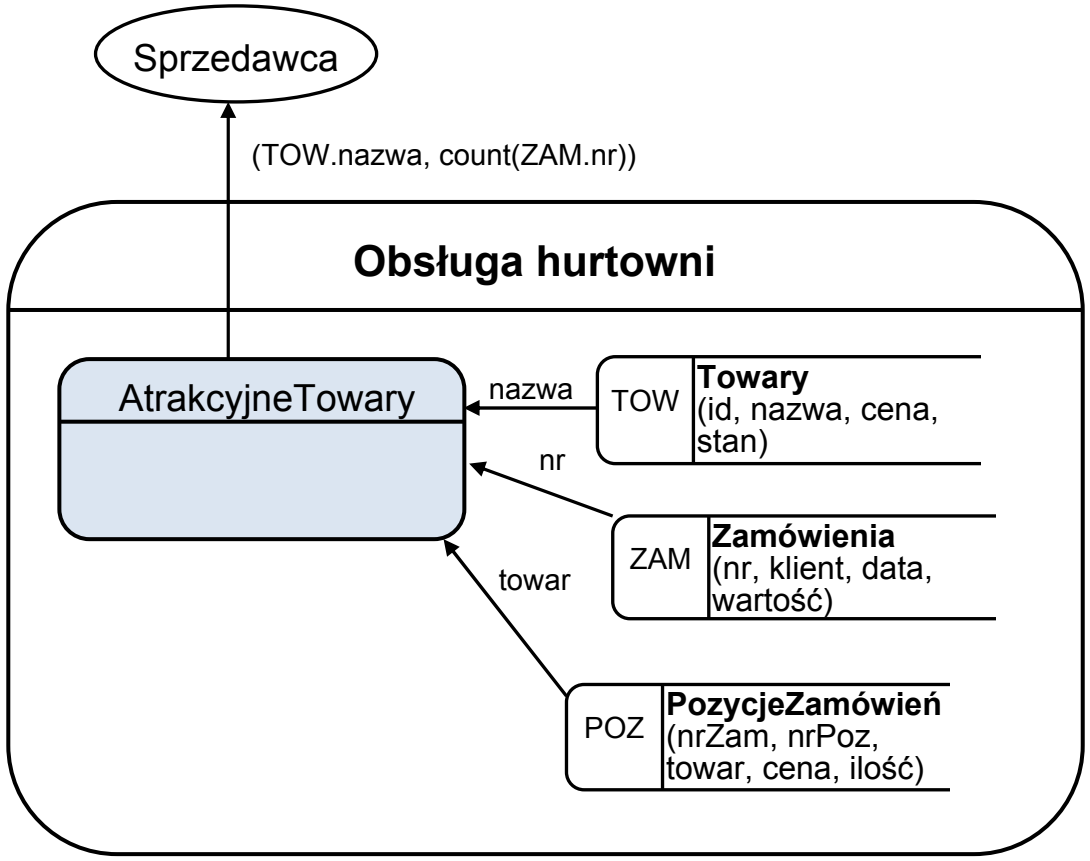


Wyznaczanie złożoności *wyjść danych*.

FTR	DET		
	1 - 5	6 - 19	> 19
< 2	Niska(4)	Niska(4)	Średnia(5)
2 lub 3	Niska(4)	Średnia(5)	Wysoka(7)
> 3	Średnia(5)	Wysoka(7)	Wysoka(7)

Wyjścia danych

Przykład: funkcja wyznaczania najczęściej kupowanych towarów.

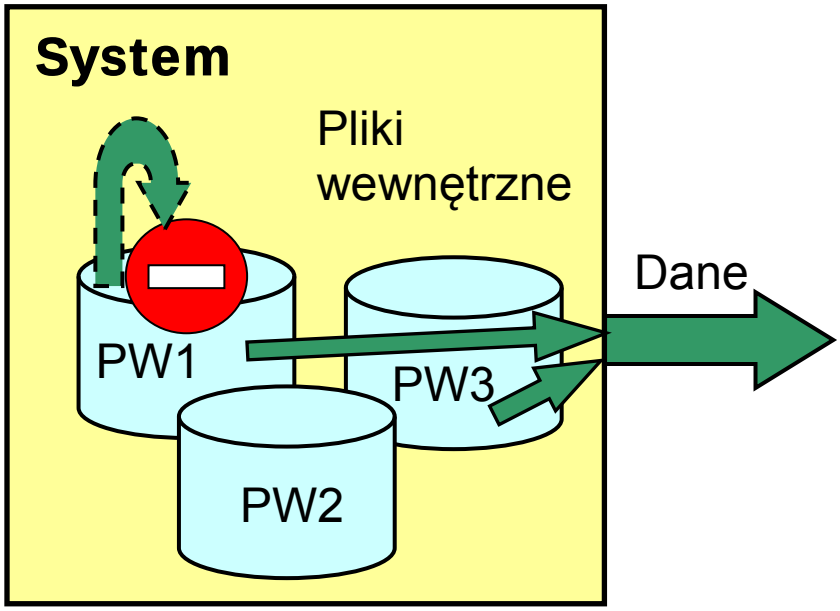


Typ funkcjonalności: wyjście danych
FTR: 2 (TOW, ZAM+POZ)
DET: 2 (TOW.nazwa, count(ZAM.nr))

FTR	DET		
	1-5	6-19	> 19
< 2	Niska(4)	Niska(4)	Średnia(5)
2 lub 3	Niska(4)	Średnia(5)	Wysoka(7)
> 3	Średnia(5)	Wysoka(7)	Wysoka(7)

Zapytania

Zapytania jest elementarnym przepływem danych, w którym dane przekraczają granice systemu na zewnątrz. Operacje zapytania nie modyfikują wewnętrznych plików danych. Dane wchodzące do systemu są parametrami wejściowymi dla zapytania. Operacje wyjściowe wyprowadzają z systemu nie przetworzone dane z wewnętrznych lub zewnętrznych plików danych

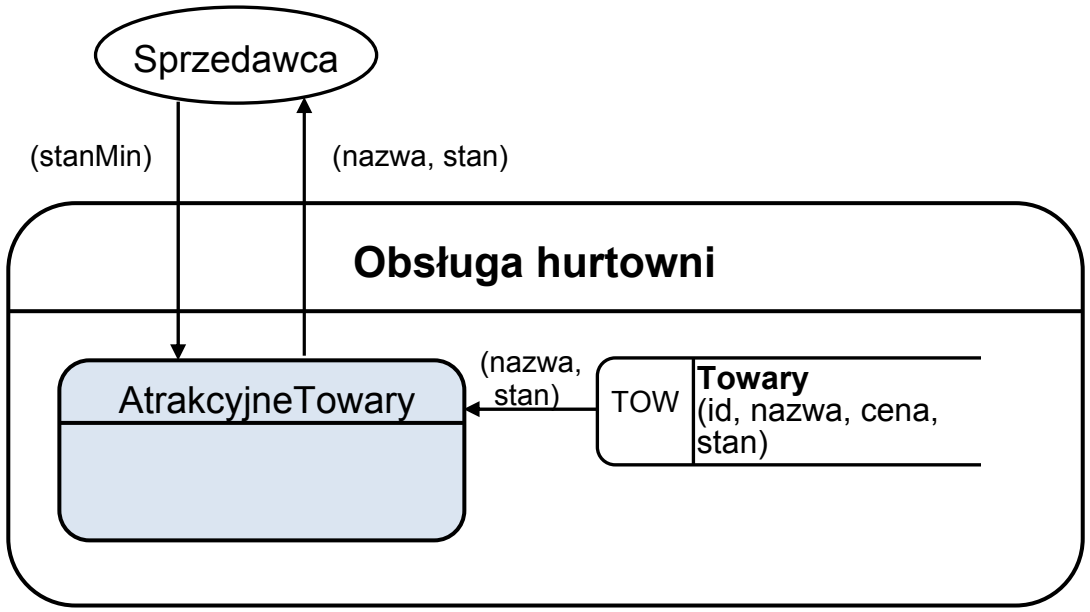


Wyznaczanie złożoności *zapytań*.

FTR	DET		
	1 - 5	6 - 19	> 19
< 2	Niska(3)	Niska(3)	Średnia(4)
2 lub 3	Niska(3)	Średnia(4)	Wysoka(6)
> 3	Średnia(4)	Wysoka(6)	Wysoka(6)

Zapytania

Przykład: funkcja raportowania informacji o towarach, których stan jest mniejszy od zadanej wartości granicznej.



Typ funkcjonalności: zapytanie
FTR: 1 (TOW)
DET: 3 (stanMin, nazwa, stan)

FTR	DET		
	1-5	6-19	> 19
< 2	Niska(3)	Niska(3)	Średnia(4)
2 lub 3	Niska(3)	Średnia(4)	Wysoka(6)
> 3	Średnia(4)	Wysoka(6)	Wysoka(6)

Zestawienie funkcjonalności EI, EO i EQ

Główne różnice między poszczególnymi funkcjonalnościami przepływu danych dotyczą podstawowego celu tych funkcjonalności.

Własności	**EI**	**EO**	**EQ**
Zmienia zachowanie systemu	**C**	**O**	**N/A**
Modyfikuje jeden lub więcej ILF	**C**	**O**	**N/A**
Przedstawia informacje użytkownikowi	**O**	**C**	**C**
Wykonuje złożone przetwarzanie danych	**O**	**O**	**N/A**

C – główny cel funkcjonalności

O – opcjonalny, pomocniczy cel funkcji

N/A – funkcjonalność nie posiada tego typu własności

Wewnętrzne pliki danych

Wewnętrzny plik danych jest identyfikowalnym przez użytkownika logicznie powiązanym zbiorem danych, który znajduje się całkowicie w granicach systemu i jest zasilany poprzez wejścia danych.

Szacowanie złożoności *wewnętrznych plików danych*.

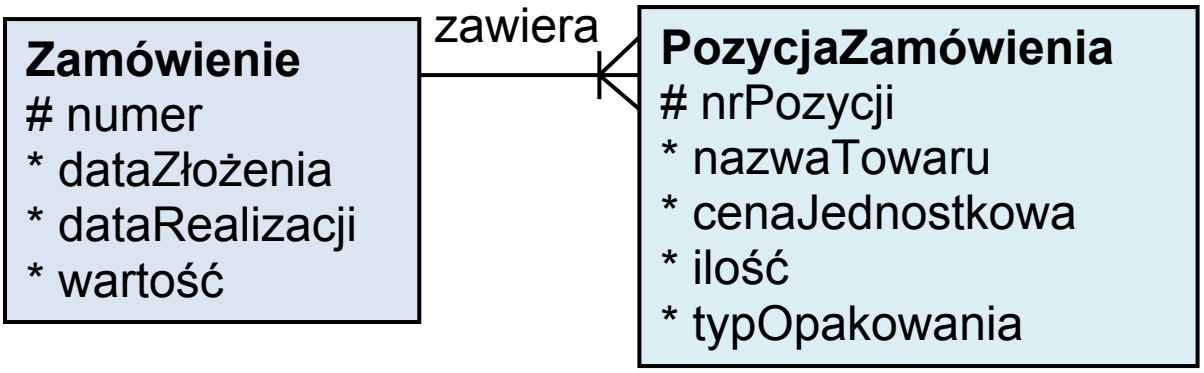
RET	**DET**		
	1 - 19	20 - 50	> 51
1	Niska(7)	Niska(7)	Średnia(10)
2 do 5	Niska(7)	Średnia(10)	Wysoka(15)
> 6	Średnia(10)	Wysoka(15)	Wysoka(15)

Ustalenie liczby struktur danych w pliku:

- Związki kompozycji określające zależność i wyłączność danych składowych
- Hierarchie encji
- Encje połączeniowe

Wewnętrzne pliki danych

Przykład: plik logiczny zawierający informacje o złożonych zamówieniach.



Typ funkcjonalności: wewnętrzny plik danych
RET: 2 (Zamówienie, PozycjaZamówienia)
DET: 10 (numer, dataZłożenia, dataRealizacji, wartość, nr pozycji, nazwaTowaru, cenaJednostkowa, ilość, typOpakowania, *zawiera*)

RET	DET		
	1 - 19	20 - 50	> 51
1	Niska(7)	Niska(7)	Średnia(10)
2 do 5	Niska(7)	Średnia(10)	Wysoka(15)
> 6	Średnia(10)	Wysoka(15)	Wysoka(15)

Zewnętrzne pliki danych

Zewnętrzny plik danych jest identyfikowalnym przez użytkownika logicznie powiązanym zbiorem danych, który znajduje się całkowicie poza granicami systemu. Dane te mogą być jedynie odczytywane. Zewnętrzny plik danych danego systemu jest plikiem wewnętrznym innego systemu.

Szacowanie złożoności *zewnętrznych plików danych*.

RET	**DET**		
	1 - 19	20 - 50	> 51
1	Niska(5)	Niska(5)	Średnia(7)
2 do 5	Niska(5)	Średnia(7)	Wysoka(10)
> 6	Średnia(7)	Wysoka(10)	Wysoka(10)

Jak wyznaczać typy i złożoność funkcjonalności

Faza strategii

- Diagram Use Case – przypadki użycia reprezentują przepływy danych. Możliwa jest zgrubna ocena ich złożoności. Przyjmuje się, że typowe przypadki użycia mają złożoność funkcyjną równą 35.

Faza analizy systemowej

- Hierarchia funkcji – elementarne funkcje reprezentują funkcjonalność przepływów danych
- Diagram encji-związków – encje reprezentują logiczne pliki danych. Związki między encjami są liczone jako elementy DET plików danych.
- Diagram przepływu danych – pozwala określić funkcje przepływu danych przez granice systemu, zbiorniki danych i złożoność przepływów danych.
- Transakcje bazy danych – pozwalają zweryfikować elementarne przepływy danych.

Faza projektowania

- Ekrany, raporty i pliki wsadowe mogą być podstawą do zliczania punktów funkcyjnych
- Schemat bazy danych – schematy relacji mogą być podstawą do wyznaczania logicznych plików danych. Klucze obce będące częścią kluczy podstawowych mogą wskazywać na: encje słabe, encje połączeniowe i hierarchie encji.

Proces liczenia punktów funkcyjnych

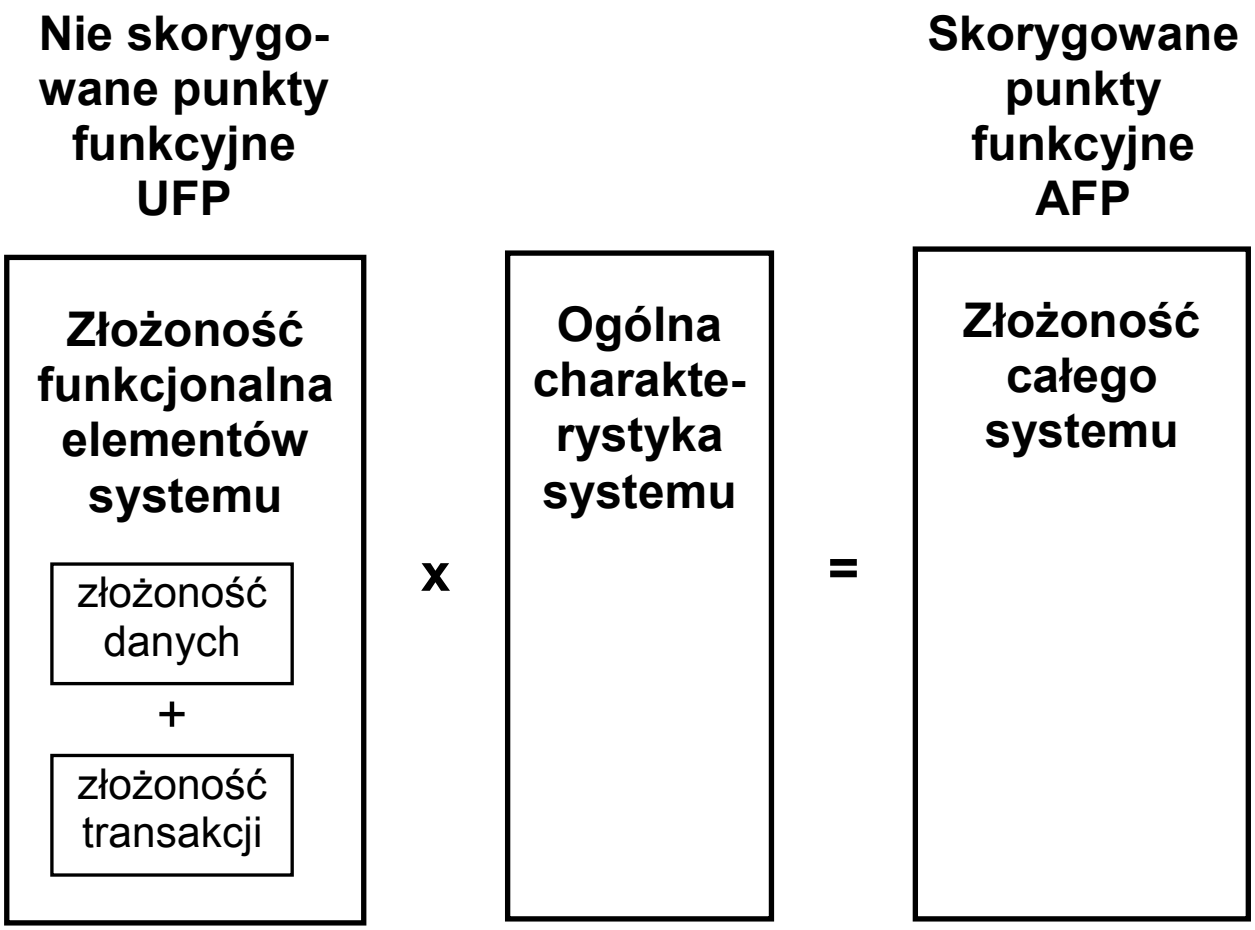
1. Identyfikacja wszystkich elementów funkcjonalnych w wyniku dekompozycji systemu informatycznego
2. Identyfikacja typu każdego przepływu danych
3. Określenie liczby dostępów do danych i liczby atrybutów przepływu dla każdego przepływu danych
4. Określenie na tej podstawie klasy złożoności przepływu danych i przypisanie na tej podstawie odpowiedniej liczby punktów funkcyjnych
5. Identyfikacja typu dla każdego pliku danych
6. Określenie liczby struktur danych i ich atrybutów dla każdego pliku danych
7. Określenie na tej podstawie klasy złożoności pliku danych i przypisanie na tej podstawie odpowiedniej liczby punktów funkcyjnych
8. Zsumowanie punktów funkcyjnych przepływów i plików danych
9. Określenie charakterystyki systemu informatycznego według 14 własności
10. Wyliczenie czynnika korygującego
11. Wyliczenie skorygowanej sumy punktów funkcyjnych

Sumowanie punktów funkcyjnych

Moduł	Wejścia			Suma
	niska	średnia	wysoka	
	3	4	6	
Moduł 1	5	12	6	**99**
Moduł 2	1	8	3	**53**
Razem	**18**	**80**	**54**	**152**
Moduł	Wyjścia			Suma
	niska	średnia	wysoka	
	4	5	7	
Moduł 1	2	15	9	**146**
Moduł 2	1	10	2	**68**
Razem	**12**	**125**	**77**	**214**
Moduł	Zapytania			Suma
	niska	średnia	wysoka	
	3	4	6	
Moduł 1	3	5	1	**35**
Moduł 2	1	4	1	**25**
Razem	**12**	**36**	**12**	**60**
Moduł	Pliki wewnętrzne			Suma
	niska	średnia	wysoka	
	7	10	15	
Moduł 1	1	3	2	**67**
Moduł 2	1	2	1	**42**
Razem	**14**	**50**	**45**	**109**
Moduł	Pliki zewnętrzne			Suma
	niska	średnia	wysoka	
	5	7	10	
Moduł 1	0	1	1	**17**
Moduł 2	0	0	0	**0**
Razem	**0**	**7**	**10**	**17**
Suma nieskorygowanych FP				**552**

Korekcja punktów funkcyjnych

Złożoność poszczególnych funkcji systemu musi być skorygowana o złożoność funkcjonalną całego systemu



AFP = UFP ± 35%

Korekcja punktów funkcyjnych

Lp.	Charakterystyka	Opis
1.	Komunikacja	Liczba elementów transferu i wymiany danych
2.	Rozproszenie	Stopień rozproszenia przetwarzania i danych
3.	Wydajność	Wymagania na przepustowość i czas odpowiedzi
4.	Obciążenie	Obciążenie docelowej platformy sprzętowo-programowej
5.	Częstość transakcji	Częstość uruchamiania transakcji
6.	Wejście on-line	Procent informacji wprowadzanych w trybie on-line
7.	Efektywność	Efektywność aplikacji użytkownika końcowego
8.	Modyfikacje on-line	Liczba danych modyfikowanych w trybie on-line
9.	Złożoność przetwarzania	Złożoność przetwarzania danych
10.	Do wielokrotnego użytku	Łatwość wykorzystania w nowych wdrożeniach
11.	Łatwość instalacji	Łatwość instalacji systemu
12.	Łatwość utrzymywania	Stopień zautomatyzowania procedur utrzymania
13.	Wielostanowiskowość	Specjalny projekt dla obsługi wielu stanowisk
14.	Łatwość konfiguracji	Specjalny projekt dla łatwej rekonfiguracji systemu

Przykładowe punktowanie

Wydajność

Punktowanie	Opis cechy
0	Użytkownik nie ma żadnych wymagań co do wydajności systemu.
1	Nie określono żadnych specjalnych wymagań dla zapewnienia wydajności.
2	Czasy odpowiedzi i przepustowość są krytyczne podczas godzin szczytu. Zakończenie przetwarzania musi nastąpić do początku następnego dnia.
3	Czasy odpowiedzi i przepustowość są krytyczne podczas całego dnia roboczego. Ograniczenia dotyczą zakończenia procesów komunikacyjnych z innymi systemami.
4	Dodatkowo wymagania dotyczące wydajności wymagają odrębnej analizy już podczas procesu projektowania.
5	Dodatkowo niezbędne zastosowanie narzędzi do analizy wydajności jest wymagane podczas wszystkich faz procesu projektowania.

Wzór na wyliczanie skorygowanych punktów funkcyjnych:

$$AFP = [0,65+\Sigma(\text{czynników korygujących})/100]^* UFP$$

Tabela przeliczeń FP dla wybranych języków programowania

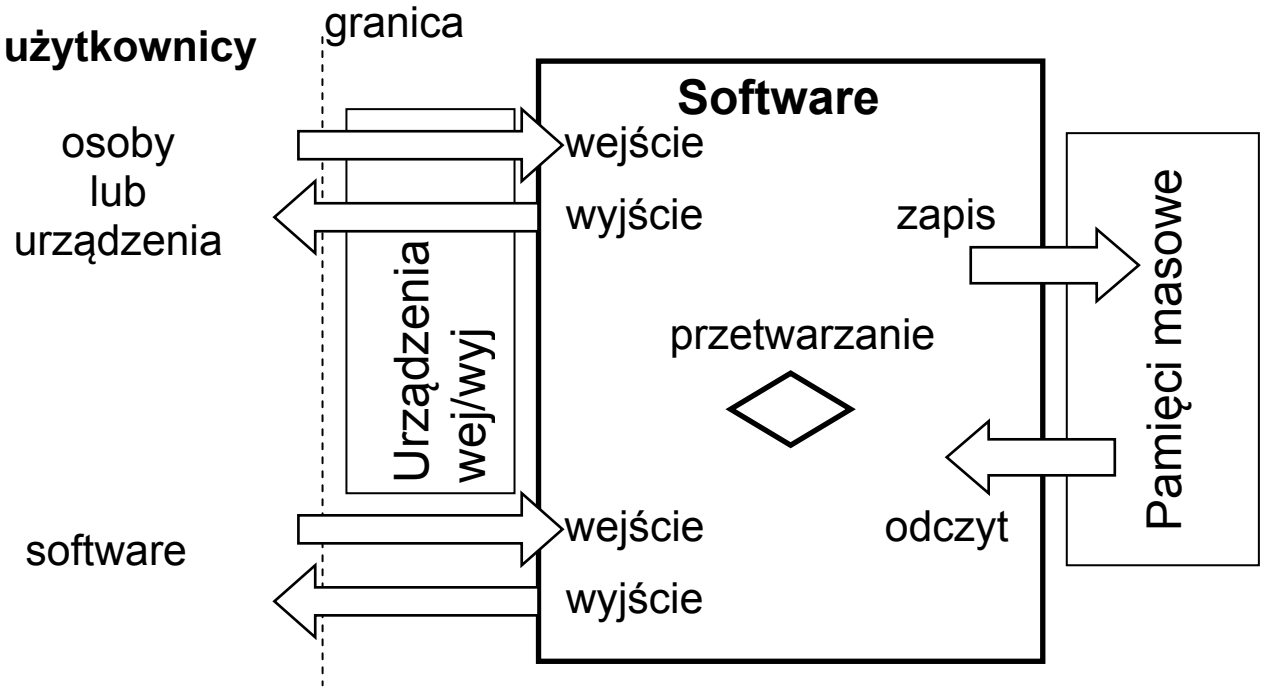
Język programowania	SLOC/FP			
	Avg	**Mediana**	**Min**	**Max**
Assembler	209	203	91	320
C	148	107	22	704
C++	59	53	20	178
C#	58	59	51	66
COBOL	80	78	8	400
Fortran	90	118	35	-
Java	55	53	9	214
JavaScript	54	55	45	63
PL/SQL	47	39	16	78
SmallTalk	28	19	17	55
SQL	31	30	13	80

Przykładowe wielkości oprogramowania

Program	Rozmiar w IFPUG 4.2	Rozmiar w SLOC	Liczba SLOC na FP
U.S. Air Traffic control	306,324	65,349,222	213
Star Wars missile defense	352,330	32,212,992	91
North Korean Border defense	273,961	25,047,859	91
DBMS Oracle	310,346	24,827,712	80
SAP	296,764	23,741,088	80
IBM MVS	104,738	11,172,000	107
Windows VISTA	157,658	10,090,080	64
Windows XP	126,788	8,114,400	64
Microsoft Office 2007 Professional	93,498	5,983,891	64
NASA Hubble controls	21,632	1,977,754	91
Patriot missile controls	15,392	1,407,279	91
Google search engine	18,640	1,192,958	64
Skype	21,202	1,130,759	53
EBAY transaction controls	16,233	742,072	46
Linux	17,505	700,205	40
NVIDIA graphics card	3,573	571,637	160
Oracle CRM Features	6,386	510,878	80
Second Life web site	14,956	398,828	27
Computer BIOS	857	274,243	320
Microsoft Excel 2007	3,969	254,006	64
Patent data mining	4,751	253,400	53

COSMIC FFP

Model funkcjonalności systemów informatycznych



Typy funkcjonalności

1. Elementarne przepływy danych
   - 1.1. Wejście (Entry)
   - 1.2. Wyjście (Exit)
   - 1.3. Zapis (Read)
   - 1.4. Odczyt (Write)
2. Przetwarzanie danych

Wersja 3 Cosmic FFP nie obejmuje funkcjonalności przetwarzania danych.

Obliczanie wielkości funkcjonalności

Faza strategii:

- Definicja celów pomiaru
- Definicja zakresu pomiaru
- Identyfikacja użytkowników funkcjonalności
- Identyfikacja poziomu szczegółowości

Faza odwzorowania

- Identyfikacja procesów funkcjonalnych
- Identyfikacja grup danych

Faza pomiaru

- Identyfikacja elementarnych przepływów danych
- Sumowanie elementarnych przepływów danych

Jedna jednostka rozmiaru funkcjonalnego (Full Function Point – FFP) jest definiowana jako jeden elementarny ruch danych dotyczący pojedynczej grupy danych.

Rozmiar funkcjonalny systemu informatycznego jest sumą rozmiarów jego funkcji składowych:

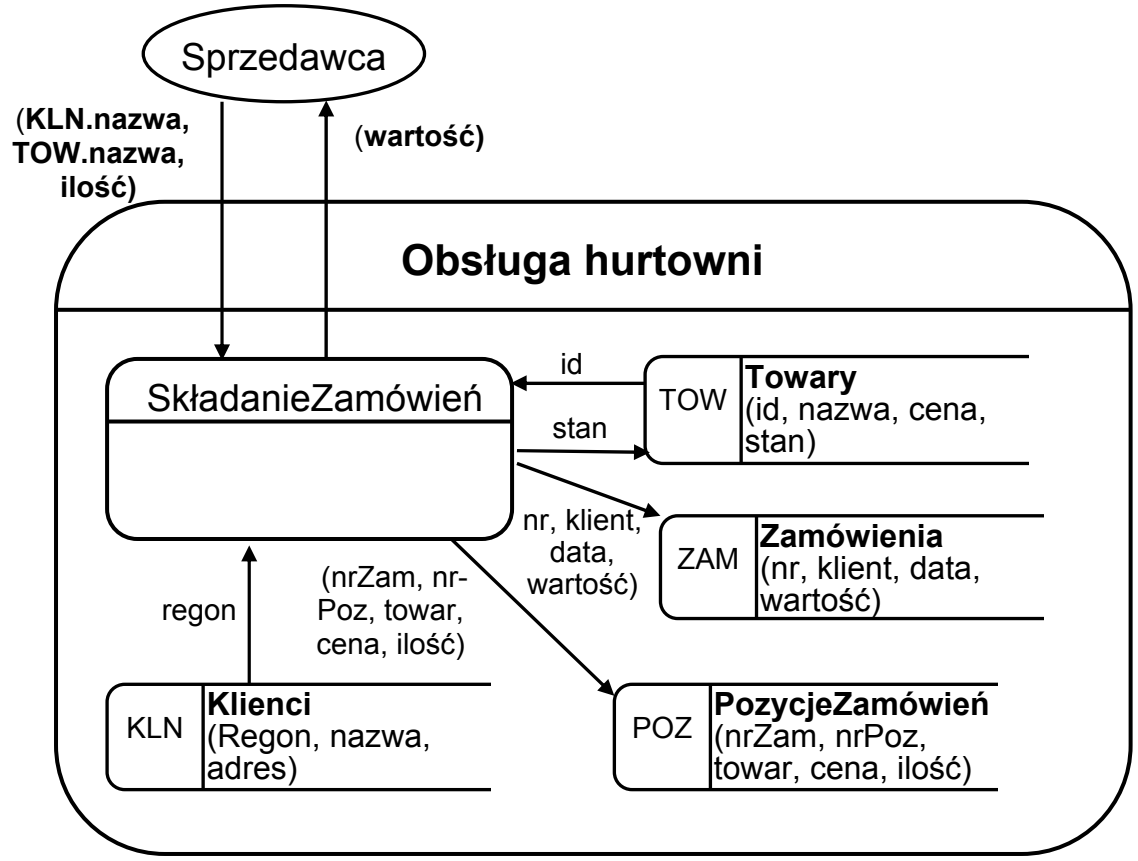
$$\mathbf{Size_{FFP}(SI) = \Sigma\ size_{FFP}(proces\ funkcyjny_i)}$$

Rozmiar funkcjonalny systemu funkcji składowej jest sumą rozmiaru jej wejść, wyjść, odczytów i zapisów:

$$\begin{aligned}\mathbf{Size_{FFP}(proces\ funkcyjny_i)} &= \mathbf{size_{FFP}(wejść_i)} \\ &+ \mathbf{size_{FFP}(wyjść_i)} \\ &+ \mathbf{size_{FFP}(odczytów_i)} \\ &+ \mathbf{size_{FFP}(zapisów_i)}\end{aligned}$$

Zliczanie FFP

Przykład: funkcja wprowadzania danych o zamówieniach.



	sprzedawca	KLN	TOW	ZAM	POZ
Składanie zamówień	**E, X**	**R**	**R, W**	**W**	**W**

1*E + 1*X + 2*R + 3*W = 7 FFP